

[useit.com](#) → [Alertbox](#) → Jan. 2005 Advanced Hypertext

[Jakob Nielsen's Alertbox](#), January 3, 2005:

Reviving Advanced Hypertext

Summary:

To manage a huge, worldwide information space, users need proven features like fat links, typed links, integrated search and browsing, overview maps, big-screen designs, and physical hypertext.

Tim Berners-Lee's genius in inventing the Web in 1991 was to strip the hypertext concept to the bone and design a system with minimal features that worked across the Internet.

The Web really has only one feature: **uni-directional plain links** that replace the existing page with a new one. Yes, the feature has twists, such as the ability to go back or open the link in a new window (which I [recommend against](#)), but fundamentally, the Web has no advanced hypertext capabilities.

New Web Features

Since the first hypertext systems in the 1960s, many more features have been invented and several turned out to be useful. Maybe it's time to implement some of these features for the Web.

Fat Links

Fat links are links that point to more than one page. Now that browsers like Firefox and Safari support **tabbed browsing**, it's possible to have a link that opens up multiple tabs, and thus lets users access several destinations in one click.

Although many users like tabs, I'm personally less enamored of them, possibly because I use a fairly large screen (2048 x 1536 pixels). With a big screen, it's usually better to manage multiple pages in windows rather than in tabs. Taskbar usability improves because you can see more of the window title in the buttons on the bottom of the screen. Also, a big screen lets you display multiple Web pages

simultaneously, which dramatically improves the usability of critical tasks like [collect, compare, and choose](#). Still, tabs do have their advantages, and in any case, they're only one possible implementation of fat links.

(Note: Bookmarks are a special kind of link; Firefox's ability to open all of a folder's bookmarks at once is thus a **fat bookmark**.)

Typed Links

If the user interface formally recognized different link types, it could **display and manage them differently**. The most obvious type distinction is between a website's internal links and links that point to other sites.

Browsers could implement this distinction today by simply looking at the domain name in the URL: if you're displaying a page from foobar.com, links to foobar.com pages are internal and all others are external. A slightly more advanced system might recognize the possibility of a given company owning multiple domains.

Many website designers have attempted to design icons that **notify users about links to external sites**, but these attempts usually fail because the designs are non-standard. Jakob's Law states that users spend most of their time on *other* sites and form their expectations based on their aggregated user experience. Thus, unless all sites have the same icons (and use them consistently), it's a doomed endeavor to visualize link typing with icons, rather than to embed types at a deeper system level.

Of course, typed links could have many uses beyond the simple internal/external distinction. For example, browsers could treat destinations that require micropayment or registration differently than free links. Similarly, they could distinguish between links to arguments that support or refute a position.

Explicit Structure

Instead of treating atomic links and pages as the UI's only concepts, we could add an explicit representation of the information architecture. Opera already does this, giving users buttons to go to a site's home page, help system, category listings, and so on.

The benefit of explicit structural commands is that they **free users from slavery to individual site designs**. Users need no longer suffer under bad sites. And, even on good sites, they can simply use a standard command that's always the

same rather than spend time trying to comprehend each site's navigation. That's a primary reason for the *Back* button's popularity: it lets you avoid searching the page for a link that might accomplish the same thing.

User-Constructed Structure

A particularly interesting form of structural UI is any structure that's built by the user and added on top of existing hypertext. Annotations and guided tours are two examples.

- **Annotations** superimpose user-generated content, such as text, doodles, or links to other sites. This structure has many uses, including simply letting you post reminders to yourself about your last experience with a site.
- **Guided tours** let you collect a series of pages and subsites and combine them with additional material into a new structure that you can communicate to others. This is great for e-learning applications, but also has more pragmatic uses. For example, you might research a business purchase and send your boss a guided tour with the pros and cons of different options.

Integrated Search and Browsing

Search is one of the main ways people access the Web, and it has the huge benefit of letting them explicitly state what they want during each visit. Unfortunately, users leave all information about their current queries behind as soon as they start browsing from the SERP (search engine results page).

In 1990, Bell Communications Research's SuperBook project proved the benefits of integrating search results with navigation menus and other information space overviews. There are three basic approaches to this. The first is simply to **annotate each navigation label with the number of search hits** in the area it points to.

A second, more advanced approach would use an indication of **aggregated search relevance**. A site might, for example, emphasize an area containing one extremely relevant page over an area with ten less relevant pages. In any case, the key is to give users a prospective view of the extent to which different navigation options relate to the current query.

Also, once users arrive at a page, it's beneficial to **highlight query terms**. Doing so makes it easy for users to judge why the page was deemed relevant, and thus easier to decide whether to stay or leave. Highlighting query terms also helps users narrow their attention to the most relevant part of the page.

A third approach to integrating search and browsing is to **display higher-value advertising**. If the site knows the user's most recent query, it can display ads related to those keywords instead of more generic ads. Of course, ads on content pages will never be as successful as the same ads on a SERP, because the user's behavior has changed from seeking to reading. While on the search engine, users are looking for someplace else to go, and thus are very likely to click on any ad that promises to solve the problem inherent in the current query. On a content page, that same ad conflicts with the user's goal to process the information and possibly return to the search engine to select the next destination. Still, an ad that's relevant to the user's current problem (as indicated by recent query terms) should beat advertising chosen with less situational awareness.

Placing search ads on content pages should particularly benefit players -- like Yahoo! and MSN -- that combine search engines with networks of other services. Such sites can directly transfer their knowledge of user queries to the non-search parts of their network. Other sites might extract the user query terms from the referrer information that's usually received when visitors arrive from search engines. Or, search engines might stop passing along the current query string and start selling it as a separate data stream to destination sites.

(Once we start transferring keyword relevance to post-SERP behavior, it will be interesting to measure how fast the user's intent diverges and thus how fast the value of targeting the previous intent decays. I would not be surprised if the keywords' value evaporated within five minutes. Web users are fickle.)

Overview Maps

In several studies of pre-Web hypertexts, having an overview map of the information space improved users' performance between 12% and 41%. Knowing where you are, where you've been, and where you can go is a significant help in navigating online information.

Navigation menus and **site maps** are two common approximations of overview maps, but neither provides the full set of features that users need. Due to space constraints, navigation menus show only a limited view of users' options. Site maps don't highlight current location, partly because users must leave their current location to access the site map as a separate pageview.

Certainly, it would help if designers followed the [twenty-eight design guidelines for site map usability](#). But ultimately, designers must integrate overview diagrams with the browser to support three core features:

- **You-are-here** markers.
- **Footprints** showing where the user has been. Sites that [change the color of visited links](#) partly offer this feature, assuming their site maps use textual links. But even these sites don't provide two essential elements of footprints:
 - Marking areas that the user has visited (even if the user hasn't been on the area's main page and thus hasn't seen the link's specific URL).
 - Using differential markings that indicate the extent of the user's visit to an area (did you see all the pages in a subsite or only a few of them?).
- **Search hit density** markers, as discussed under "Integrating Search and Browsing."

Big-Screen Design

In principle, the Web is independent of screen size. But in practice, the Web is designed for a small-screen user experience, in which users view one page at a time at a narrow width (typically 800 or 1024 pixels). Scrolling is the main way users access information that can't be displayed on a small screen.

Once we get screens the size and resolution of a broadsheet newspaper, the user interface will change. It will become possible to rely more on spatial hypertext and less on linear scrolling. In fact, the very concept of a page may vanish and be replaced by higher-level aggregate units that combine multiple data feeds.

The prevalence of [portlets on intranet portals](#) is a weak precursor to the potential for integrating multiple information units that can be independently activated and updated.

Physical Hypertext

Rather than click underlined words on a screen, it's possible for users to retrieve a destination page through some real-world action. Physical objects can be the anchor points for a hypertext link.

Several doomed projects have tried to implement physical hypertext on the Web. The most prominent and clueless was CueCat: a barcode scanner that let users scan magazine ads to display more information about advertised products.

CueCat failed for two reasons:

- **It benefited advertisers, not users**, so people had little incentive for

keeping a CueCat around for those rare occasions when they wanted follow-up info on an ad. Typing in a URL is easier than digging out a special hardware device.

- **It was tied to the PC**, so it didn't help people when they were shopping or otherwise might want information but didn't have PC-based Web access.

Future projects for physical hypertext must overcome these two downsides. It will be easy to embed barcode scanners and RFID readers in mobile phones and other PDAs with Internet capability. Such devices will let users link off physical objects they encounter when they're out and about. Where should the links lead? Not just to ads, but also to other useful information, such as review sites or comparison-shopping sites that tell users whether they're getting a good deal.

Collaboration

Browsing is a solitary experience. Life isn't.

I mention collaboration as my last feature, because it's the one for which old research provides the least guidance. There's some work on shared hypertexts, where people build up a collaborative knowledge base and/or solve other problems together. Wikis offer a primitive example of the power of multi-user hypertext. But mainly, collaboration remains a field with immense promise and little progress.

Any Hope?

[In 1995, I listed fifteen hypertext features](#) that were missing from Web browsers. None of these ideas have been implemented in the ten years since, except for Firefox's search box and Internet Explorer's search sidebar.

Is there any hope that the next decade will bring more progress? I think so. For one, most of the ideas mentioned here are rich sources of user interface patents, which offer a sustainable competitive advantage. (I invented at least five potential patents while writing this article, but didn't bother filing because I'm not in the business of suing infringers; a big company could rack up the patents if so inclined.)

The last ten years were a black hole: much attention was focused on doomed attempts at making the Web more like television. Hopefully, the next decade will focus instead on empowering users and giving us the features we need to master a worldwide information space.

Learn More

For an in-depth overview of thirty years' research into hypertext theory, see my 1995 book, [Multimedia and Hypertext](#).

-
- > [Other Alertbox columns](#) (complete list)
 - > [Sign up for newsletter](#) that will notify you of new Alertboxes
-

[Copyright](#) 2005 by Jakob Nielsen